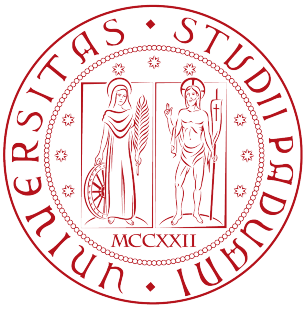
UNIVERSITÀ DEGLI STUDI DI PADOVA
DIPARTIMENTO DI FISICA ED ASTRONOMIA "Galileo Galilei"



CORSO DI LAUREA IN FISICA

Study of the production of B_s mesons in the $D_s^*\pi$ decay channel using LHCb data

Relatore: Dott. ALESSANDRO BERTOLIN

Laureando: GIULIO FAVARO

ANNO ACCADEMICO 2016/2017

Contents

Chapter 1

Introduction

The grow in power of computers and the enormous amount of data available from experiments lead in the last years to the birth of new methodologies to treat data analysis. Artificial Intelligence (AI) techniques such as neural network are now commonly used by scientist all over the world but, although wide spread, are currently not fully understood. This situation requires a study of the proprieties of the algorithm used in each specific case. In this thesis I tried to look after the techniques used in the analysis of a specific B_s^0 decay, using data from the LHCb[1] experiment operated at the LHC[2], CERN in order to investigate an enhancement of the performance of the classification of the events.

1.1 Physics case

The data used in this work are from the LHCb detector: a forward spectrometer with a single arm whose main purpose is the detection of beauty or charm mesons produced in high energy pp collisions in the pseudorapidity range $2 < \eta < 5$. We will focus on the following B_s^0 decay:

$$B_s^0 \to D_s^{*-}\pi^+ \tag{1.1}$$

Shortly after the initial B_s^0 decay the D_s^{*-} decay as:

$$D_s^{*-} \to \gamma D_s^-(\to K^+K^-\pi^-) \tag{1.2}$$

Here and in the following the charge conjugated process is always implied. This process is of physical interest because it is being used as control channel for the $B_s^0 \to K^+D_s^{*-}$ decay which was first observed in 2011/2012 LHCb data[3]. The $B_s^0 \to K^+D_s^{*-}$ channel is hoped to be used for a new measurement (or at least an independent one) of a specific CP violation parameter: the angle γ of the Cabibbo Kobayashi Maskawa matrix.

1.2 Goals

The large amount of variables that define this decay process make it difficult to separate Background from Signal because both populate complicated regions in a large multidimensional space. The AI approach is mandatory to find where the main part of the Signal lies. To obtain these results an AI technique is trained with a mixture of real data and Monte-Carlo simulations and, once fully trained, the algorithm can be applied to additional data to enhance the Signal component.
My first objective in this thesis was to take the actual ROOT based machine learning algorithm[5] used at LHCb and try to boost its performances by changing how the training is done; later I tried to compare different ROOT based methods to see which one had the best performances. Last I tried to compare the ROOT based methods with an SKLearn based application[4] to find out if more recent algorithms have better performances.

Chapter 2

Machine learning in data analysis

Modern data analysis can rely on large chunk of data and help of powerful calculators; this enhancement of the tools lead in the past years to the birth of new methodologies to treat data. Machine learning tools are the main ingredient of this new trend. Machine learning techniques focus on treating classification problems where the setup of a conventional algorithm is difficult due to the large amount of variables involved. These are particularly useful in separating Background from Signal in particle physics data analysis.

2.1 Training

The first step when aiming for an AI approach is to train the selected algorithm. This can be achieved in two main ways: by using a supervised training or by using an unsupervised one; while the former aims to teach the program to recognize a specific case the second tries to find general unknown patterns. In particle physics the common approach is to use supervised training due to the possibility to use Monte-Carlo simulation to generate reliable sets of Signal.
To start a supervised training a training set is needed. The training set is a classified sample of data which to be effective must be explicative of all the possible situations the program could encounter. An AI program commonly classifies data by associating to each event a continuous variable, usually in the range between -1 and 1, used to separate the initial sample in a Signal enriched region and a Background enriched region: a training sample must be provided for both the Signal and the Background categories and the same must be done for the testing sample. In figure 2.1 we can see an example of classified events distributions for Background and Signal.

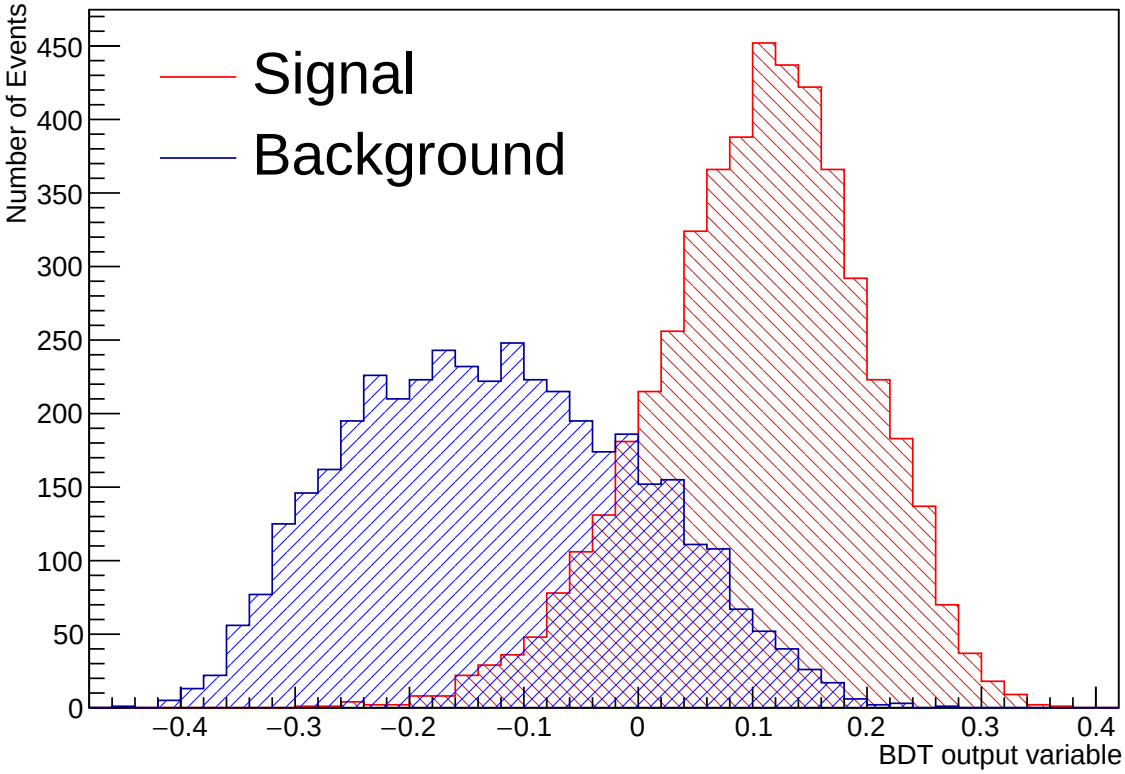


Figure 2.1: Example of a Classified Signal and Background events distributions for a BDT method.

The training sample, before the actual training start, is splitted into two parts: one actually used for the training and one used as test sample, to calculate the score of the algorithm. The train of the program is repeated with multiple iterations; each time the algorithm tries to predict the outcome of the training sample, it recognize its errors, and tries to reconfigurate itself to correct them. After each cicle the program is applied on the test sample and a score is calculated: if the score reach a fixed amount the train ends, if not the process is repeated another time; in case of over many iterations the program is unable to improve anymore the program can recognize the state as the best possible outcome and exit succesfully the training anyway.
The data used in this work are from the LHCb detector in pp collisions after various processing phases.The online event selection is performed by a trigger, which consists of a hardware stage, based on information from the calorimeter and muon systems, followed by a software stage, which applies a full event reconstruction. After the software stage and the offline event reconsruction data are finally obtained with a selection of the interesting quantities and are available as root files; this work used roughly 350.000 events with about 60 variables each identifying all the main characteristics of the events such as the momentum of the produced particles and the angle of production. Among all these variables the AI technique is fed only with the ones having the most discriminating power as cross checked by the ranking of the variables obtained from the algorithm itself.
To form the training sample in this work the Signal was taken from a Monte-Carlo simulation while the Background was taken from real data as shown in figure 2.2; in this work data were separated in multiple regions based on two variables: the mass of the B_s^0 candidate and the mass differences between the D_s^{*-} and the D_s^- candidates. The regions details can be seen in table 2.1; while the Signal was always taken in the same region Background among different ones: Default, which was used prior to this analysis, Split, that was chosen using the two regions Split A and Split B, and Vertical.

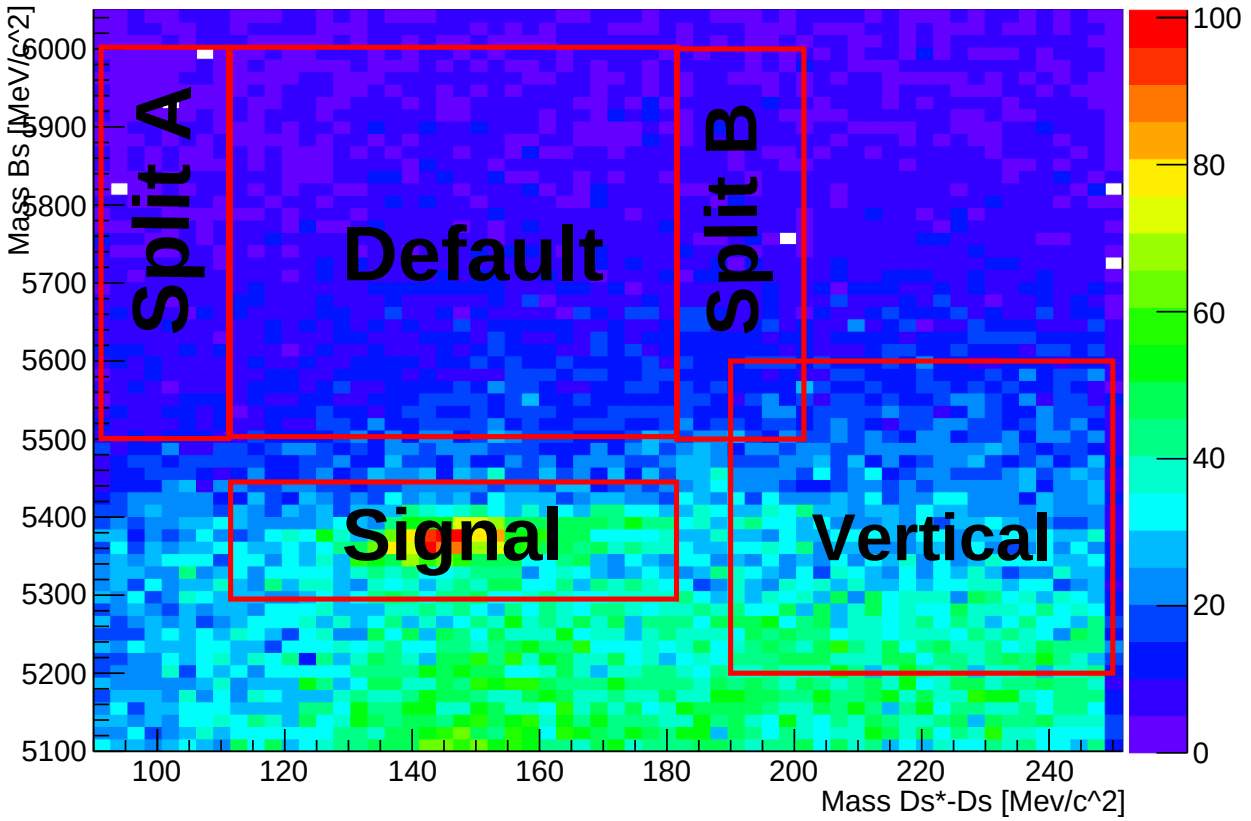


Figure 2.2: Regions of interest for Background and Signal data.

Region	B_s mass interval (MeV/c^2)	$D_s^* - D_s$ mass interval (MeV/c^2)
Signal	5295-5445	111.5-181.5
Default	5500-6000	111.5-181.5
Split A	5500-6000	191.5-111.5
Split B	5500-6000	181.5-201.5
Vertical	5200-5600	190.0-250.0

Table 2.1: Regions boundaries.

2.2 Overtraining test

A serious risk in an AI training is the overtraining phenomena. If the training is done for an excessive number of iterations or the training sample is too small there is the possibility that the algorithm reaches an ad-hoc configuration. This case is particularly dangerous cause it could go undetected by the algorithm and classified as the only good result to be found. But when used on real data the program is unable to do proper classifications. The main method to avoid overtraining is done via the test sample. The first reason to have a test sample in fact is to test the program on slightly different data that have the same proprieties of the training sample hence avoiding excessively specific configurations and this is done by comparing the distribution in the BDT output variable for Signal/Background for the training/testing sample.
Both Signal and Background distributions from training are taken and confronted with the respective ones from the test. Because they are random part of the same sample there is no reason to have different distributions. So in this work I checked the compatibility between them via two test: χ^2 test and Kolmogorov test. The output data from training and testing have been put into two histograms and the test have been run in the region where both the training and test distributions had non zero entries; the discriminant to accept the compatibility of the

distributions was set at the 95% of the confidence level. All the distributions have been found to be compatible. An example can be seen in figure 2.3

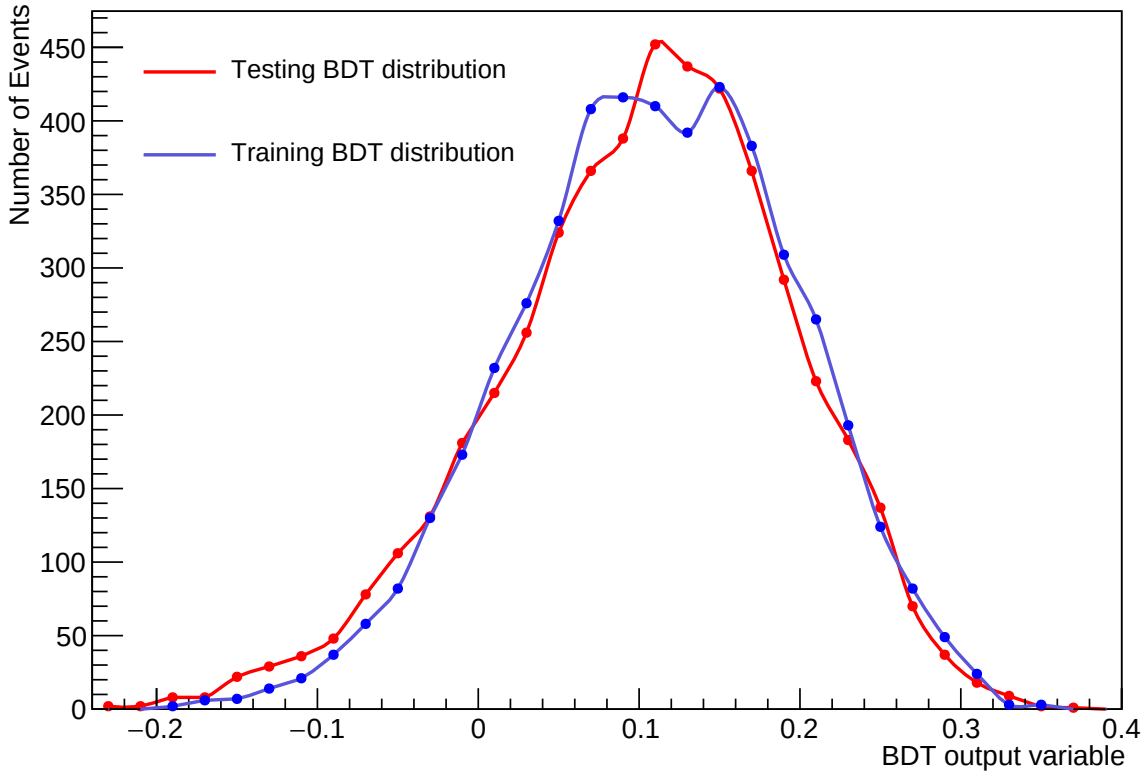


Figure 2.3: Example of a training-testing BDT output variable distribution.

2.3 Application

In this phase I took a different sample of data and run the trained AI technique on it. The final goal of this process was to create for each different tested configuration of the algorithm a ROC curve that show how much Signal we can keep while eliminating Background events. ROC curve can be used together with candidates invariant mass plots to compare different learnings

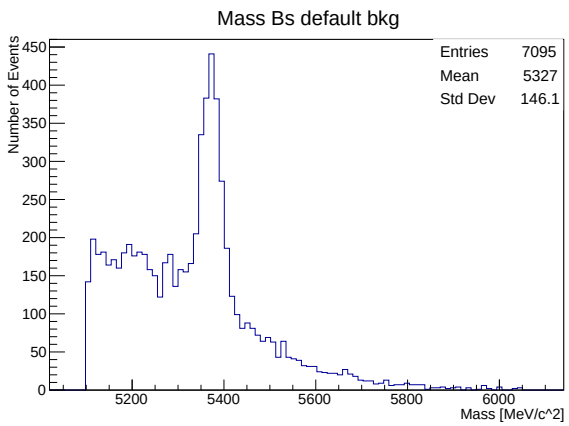
Chapter 3

ROOT based methods

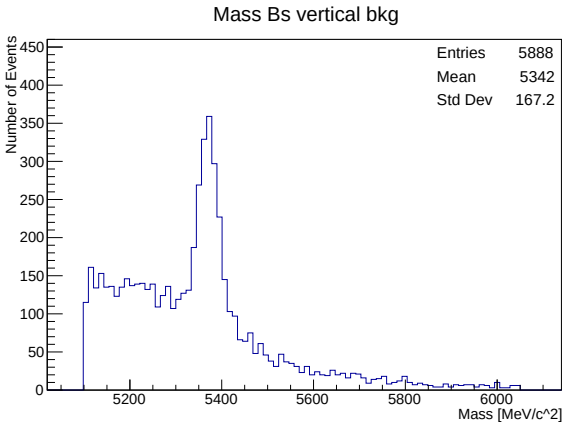
TMVA is one among the numerous toolkits available inside ROOT. TMVA provides several machine learning algorithms fully integrated in the ROOT framework and is the actual standard toolkit used at LHCb for machine learnings approaches. The first goal of this work was to learn the behavior of TMVA under different conditions and understand how good was the approach applied until now.
The Default analysis procedure rely on a Boosted Decision Tree (BDT) application trained with Background taken in the so called Default region. We started by trying to change the Background region and later focused onto different AI methods. To actually see which configurations lead to the best results we choose two main ways: the former is based onto the comparison of the associated ROC curves while the latter compare the mass plots of the particles involved in the analysis. The working point is chosen by discarding 90% of the Background events.

3.1 Background regions comparison

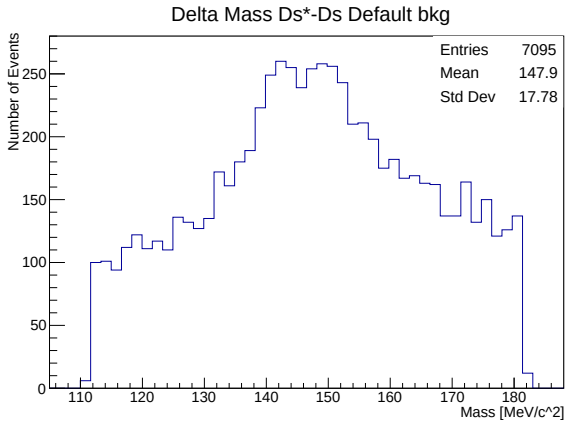
As the learning is the core of a machine learning approach we started by looking if different Background regions, independent from the Default one, could make an improvement in the algorithm performance.
We first thought that the region near the Default one could contain additional useful informations so as a first try I took a sum of the two Background regions adjacent the Default one (Split A and Split B). We observed no sensible improvement in the BDT power as can be seen in figure 3.4.
We proceed to test a "vertical" region: as the test that were done focused on a large differences in the $D_s^* - D_s$ mass spectrum but values of the B_s mass close to the nominal one, we tried to take a side region with opposite characteristics. Using the same BDT a new learning was performed but it showed poor results: in figure 3.4 we can see how the ROC is sensibly lower compared to Default one, we can see the difference in the mass plots, see figures 3.1a, 3.1b, 3.1c, 3.1d.



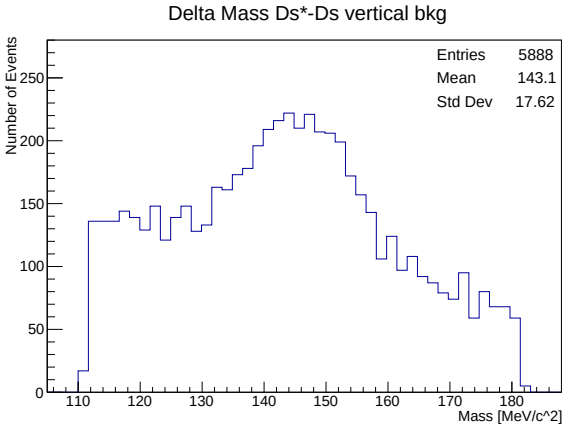
(a) Default Background B_s mass plot.



(b) Vertical Background B_s mass plot.



(c) Default Background $D_s^* - D_s$ mass difference.

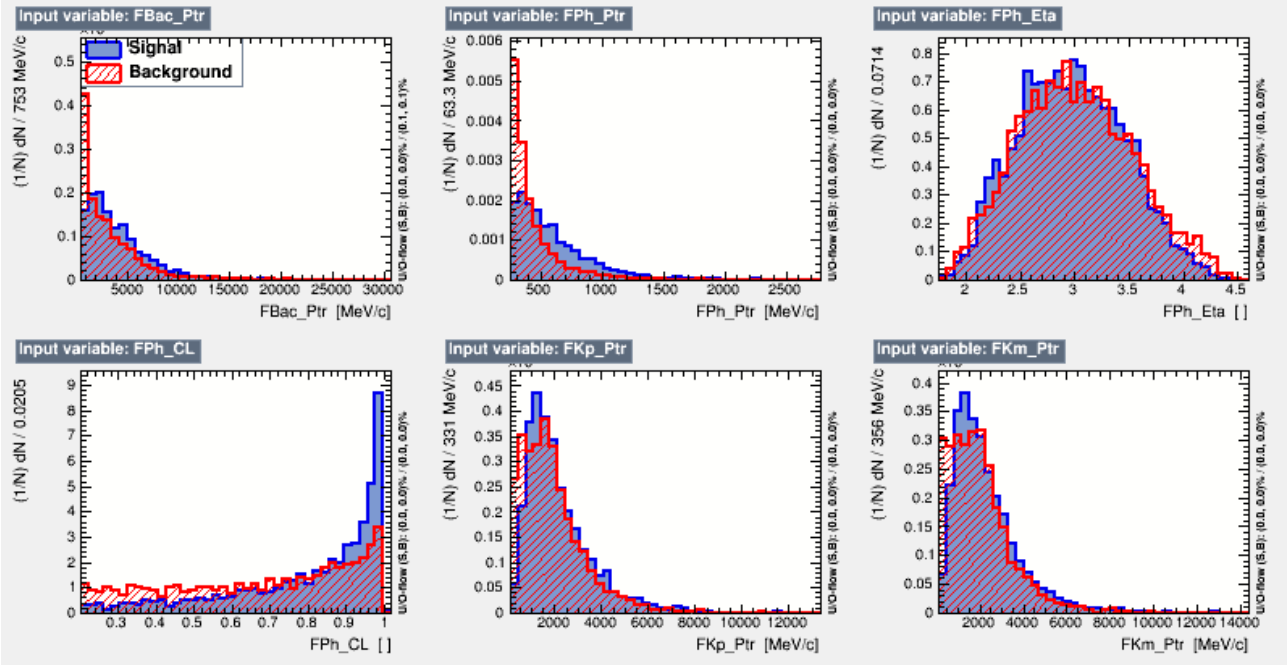


(d) Vertical Background $D_s^* - D_s$ mass difference.

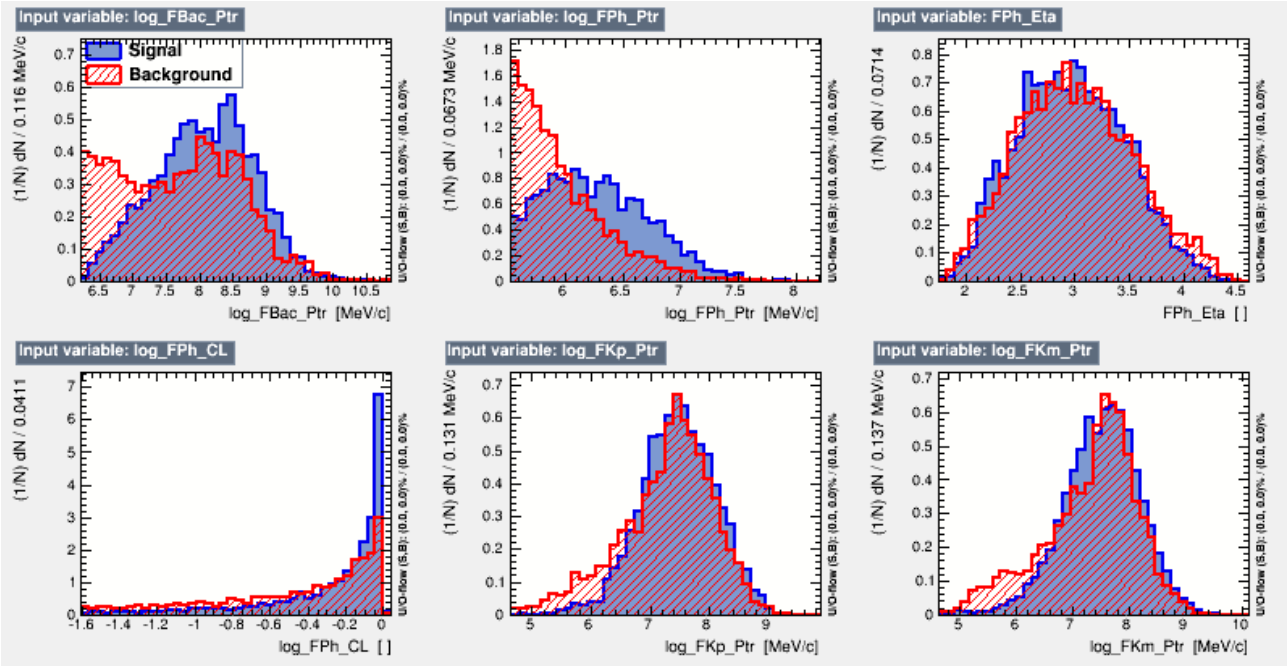
Figure 3.1: Candidates invariant mass plots for different configurations.

3.2 Algorithm comparison

Instead of changing the Background region we later focused on changing the way to analyze the data.
We started by searching how to have more "Gaussian" input variables hence we took the logarithm of all input quantities used in the Default training. An example of the transformations that the distributions had under logarithm is shown in figure 3.2 while in figure 3.3 we can see the comparison between the ROCs curve for logarithmic and standard variables.



(a) Default distributions.



(b) Logarithmic distributions.

Figure 3.2: Confront between variables distributions before and after logarithm.

As in TMVA many algorithms are present I tried to use a different one and look for differences. The Fisher method, based on the Fisher linear discriminant, was tried with the Default Background region. Mostly because no sensible differences were found for other Background choices. The overall score of the method was worse than any other, as seen in figure 3.4 so any further test was abandoned.

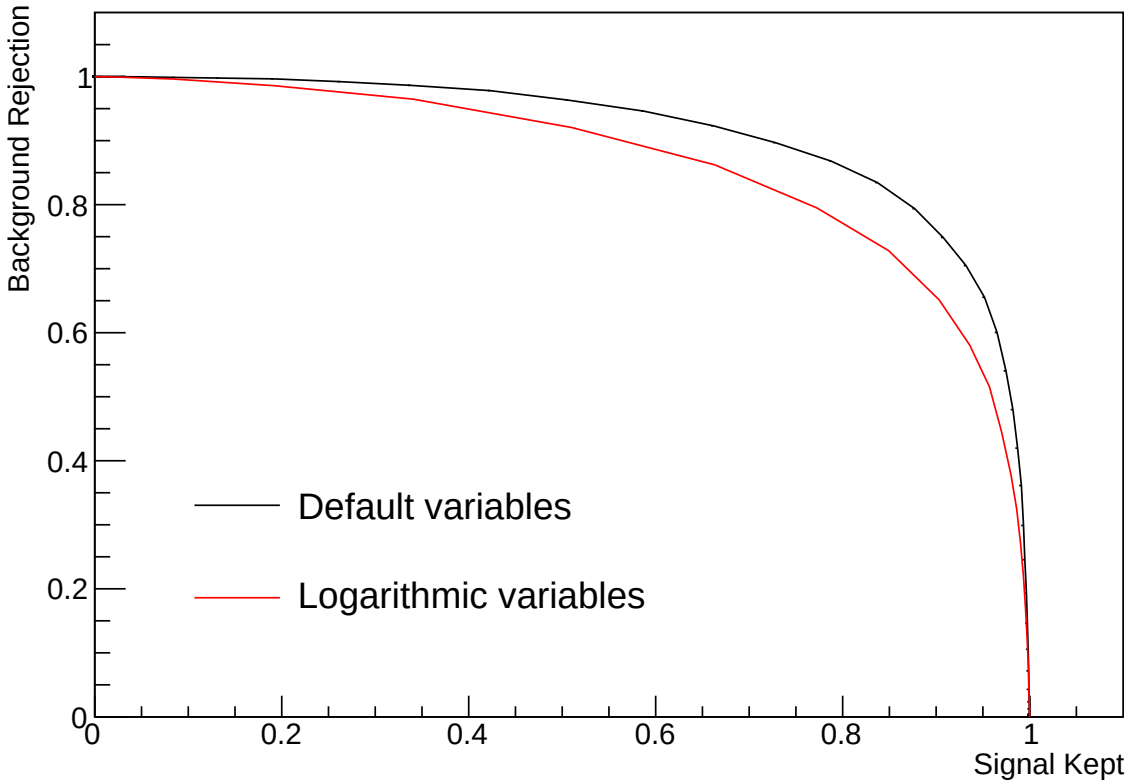


Figure 3.3: ROC curves: logarithmic vs standard variables.

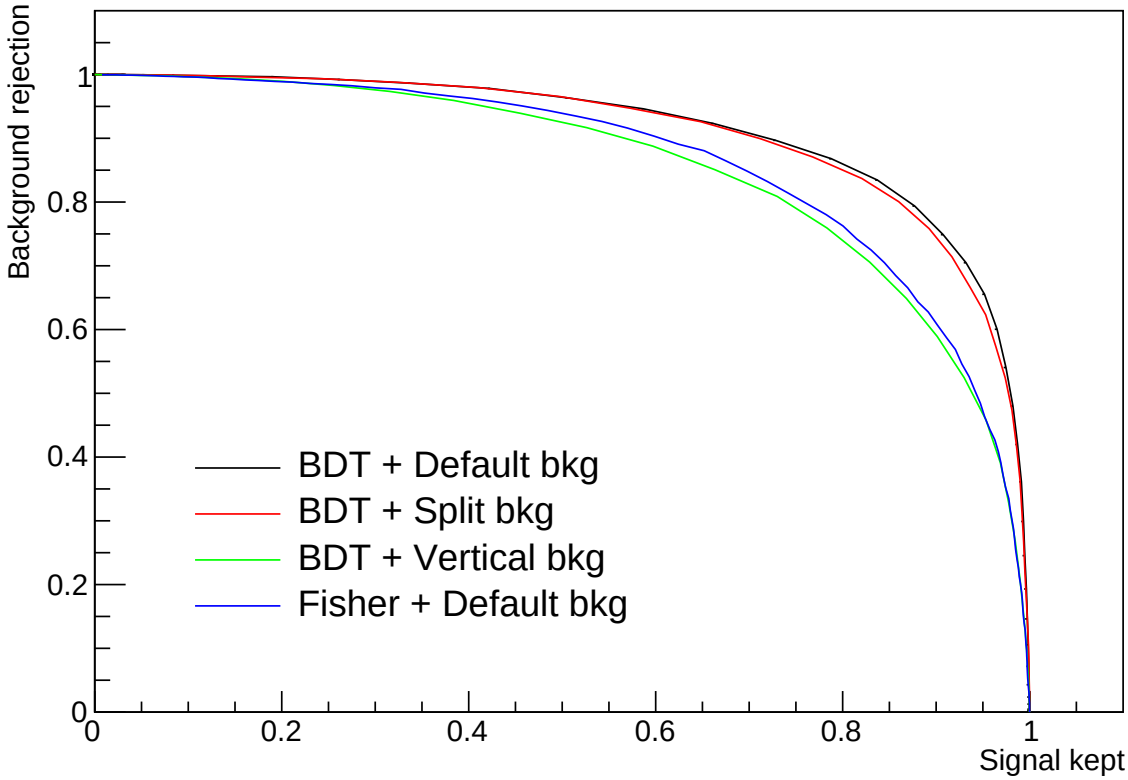


Figure 3.4: ROC comparison between different Background training regions and a Fisher based method.

3.3 Sample size comparison

The learning process is obviously influenced by the number of events available in the training sample but a too large sample isn't preferable either: while more events cover a bigger spectrum of possibilities, hence avoiding overtraining, a big sample leads to a sensible increase in CPU training time.
Basically we mostly wanted to discover if the actual analysis was already in an asymptotic region or if there was room for improvement. I took here the Background Default region and built several samples with increasing number of Background events ranging from 100 to 8000 (we found that under 100 events the BDT wasn't able to complete the training). For each test an equal amount of Signal event was added from MC.
The BDT was trained using these different sized samples and the results compared. While a dependencies from the sample size was present, after 2000 events there was almost no differences between the trials. Results are shown in figures 3.5 and 3.6.

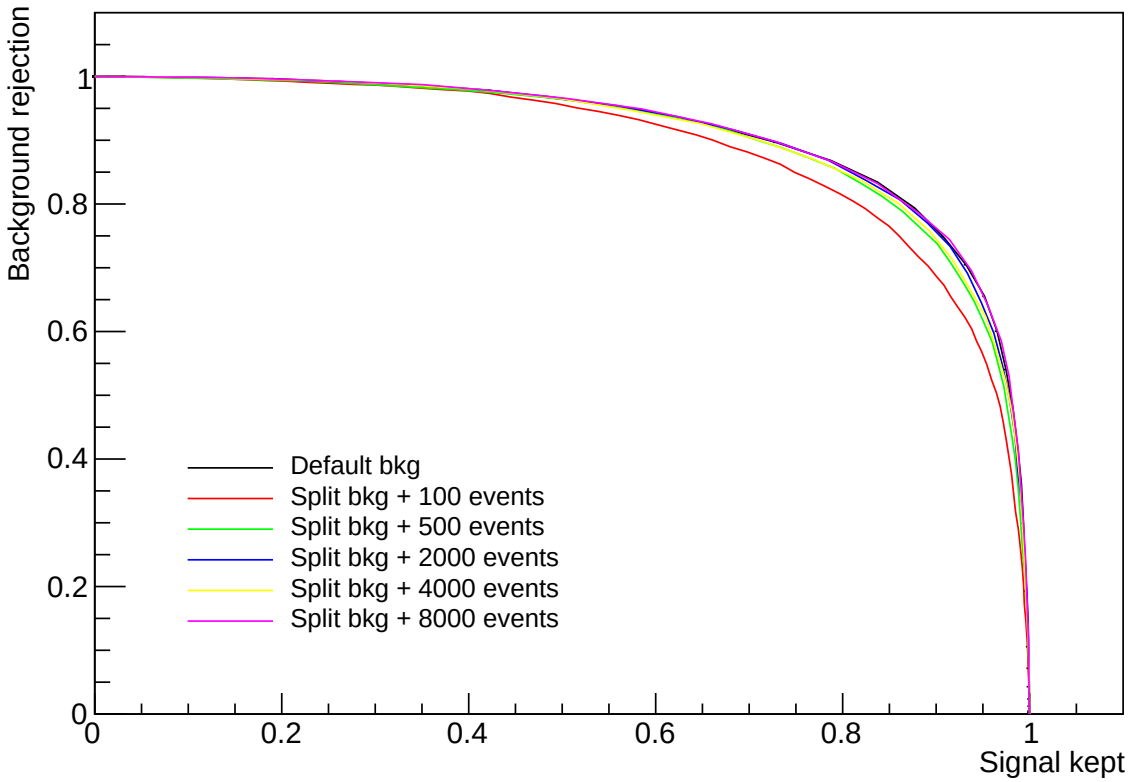


Figure 3.5: ROC comparison between progressively larger Signal and Background training samples.

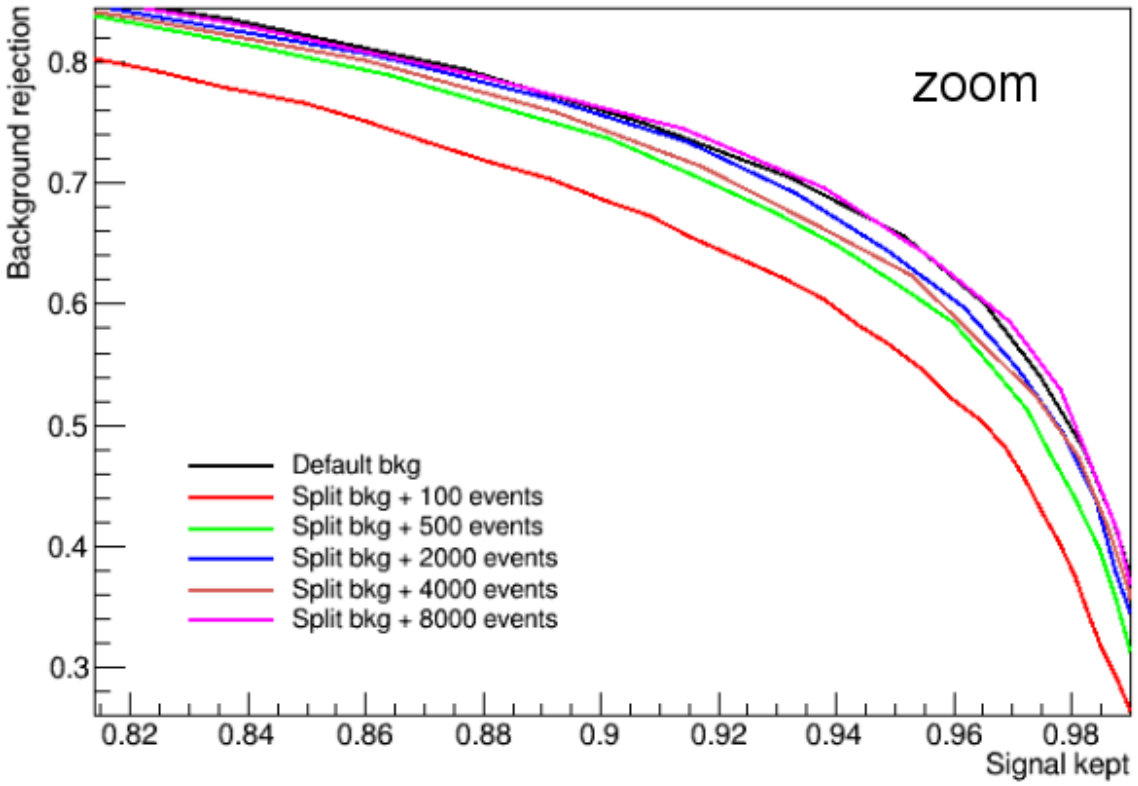


Figure 3.6: Particular of the region of interest.

Chapter 4

SKLearn based methods

One of the goals of this thesis was to test more recently developed machine learning techniques to see if they could lead to a boost of the performances. The SKLearn python library was a good candidate for this. The first obstacle to overcome was the need to merge the plain C++ framework of the LHCb analysis with the python framework used by SKLearn. This was achieved using a tool already developed in LHCb[6].
The training under SMVA followed the same steps used under TMVA. We chose to use an ADABoost algorithm, an enhanced version of the BDT algorithm, and test the performances using the two Background regions that showed the best Signal to Background separation.
As we see in figure 4.1 the two configurations lead to the same results.
Figure 4.1 also shows the comparison between the TMVA Default version and the corresponding SMVA configuration: no sensible improvement was found.

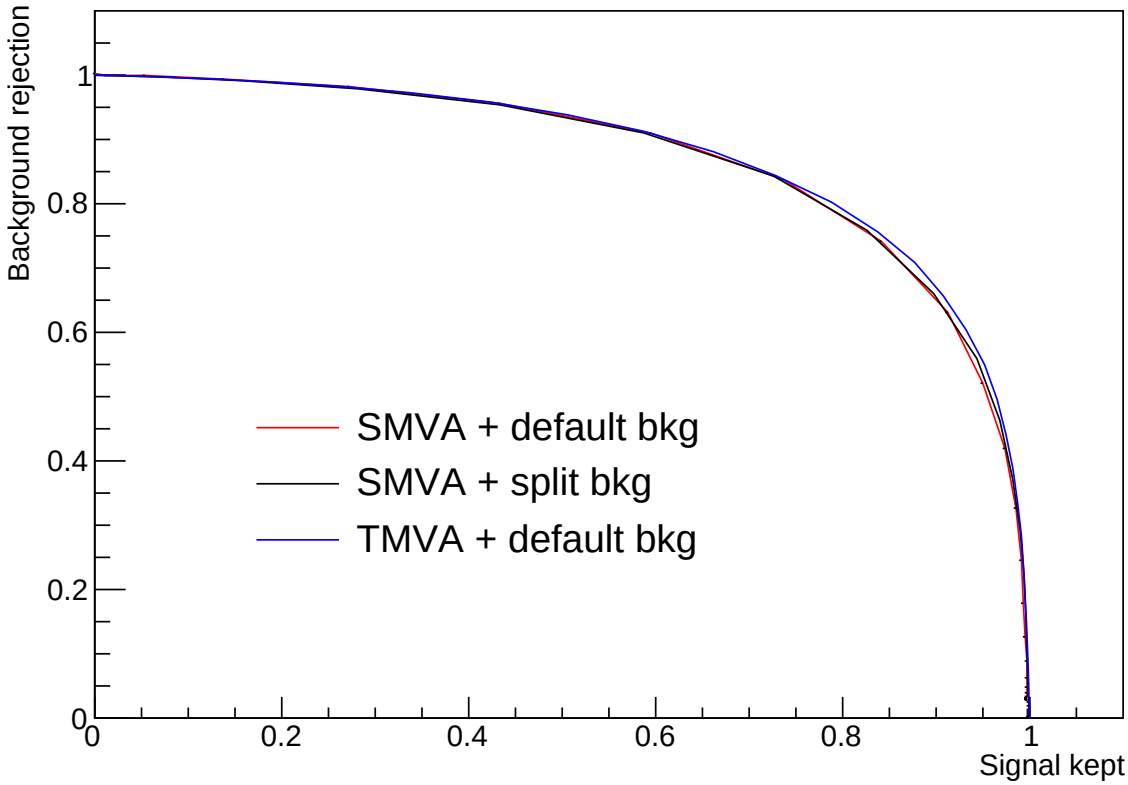
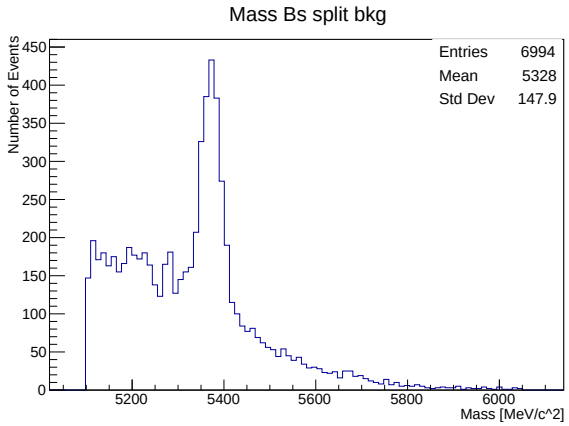
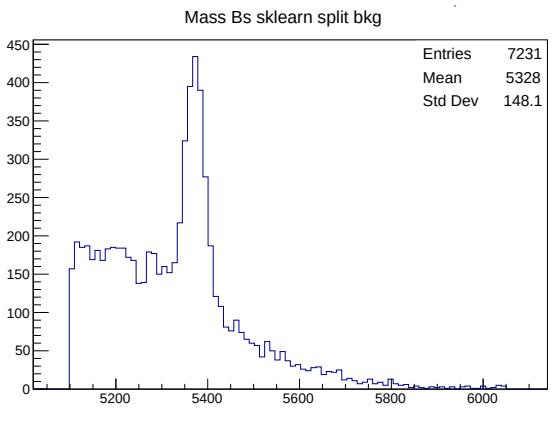


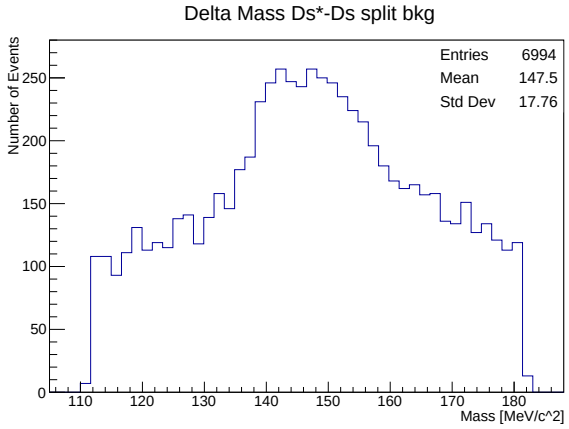
Figure 4.1: ROC curves for SMVA based algorithm vs TMVA Default configuration.



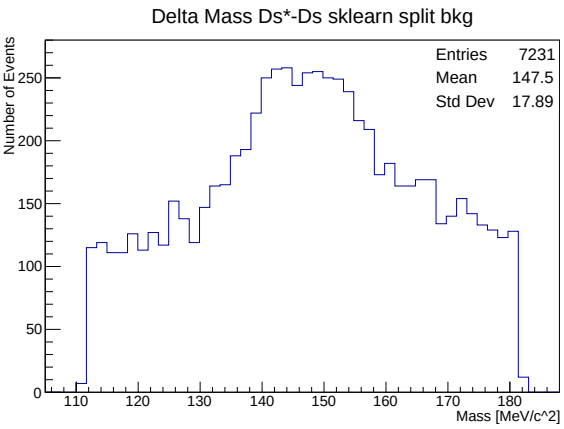
(a) TMVA Split Background B_s mass plot.



(b) SKLearn Split Background B_s mass plot.



(c) TMVA Split Background mass $D_s^* - D_s$ difference.



(d) SKLearn Split Background mass $D_s^* - D_s$ difference.

Figure 4.2: Comparison of TMVA BDT and SKLearn ADABoost methods.

Chapter 5

Global comparison

Tab 5.1 summarizes all the trials done in this work. For each configuration we report the working point of the discriminant variable used, the number of events rejected and the percentage of Signal events passing such selection (as can be seen the Background rejection is approximately constant at 90%).
While it is possible to find some configurations that work in a way sensibly worse than the others there is no clear winner among the rest of the probed configurations.

Method	Working point	Rejected Background events	Signal kept
BDT + Default	0.042	6414/7131	72.22%
BDT + Split bkg	0.08	6414/7131	71.18%
BDT + vertical Bkg	0.047	6417/7131	57.00%
BDT + Split 100 events	0.179	6416/7131	66.20%
BDT + Split 500 events	0.075	6413/7131	71.12%
BDT + Split 2000 events	0.044	6409/7131	72.55%
BDT + Split 8000 events	0.021	6413/7131	72.41%
BDT + Logarithmic variables	0.1915	6414/7131	57.64%
Fisher + Default	0.19	6417/7131	65.08%
sklearn + Default	0.5046	6410/7131	72.00%
sklearn + Split bkg	0.5082	6410/7131	71.84%

Table 5.1: Final comparison between configurations.

Chapter 6

Conclusions

The objective of this work was to investigate the actual way of selecting $D_s^*\pi$ data in LHCb by searching possible improvement. A final consideration we can state that there are no motivation to change the actual setup. In fact while several variants of the standard configurations were tried no one showed significantly better performances; different algorithms also showed no differences in performances.
The lack of differences between a lot of methods could indicate that we entered an asintotic regions were no major improvement are no more possible, this is supported by the shape of the ROC curves, at high values of Signal efficiency for a broad range of Background efficiencies, very high showing that we already hit a considerable precision in separating Background from Signal.
Although this thesis work is over there are still some possible investigations to be done: under TMVA several algorithm remain to be tested and more Background regions could also be tried. Further studies could also investigate the change in performances when using logarithmic input variables maybe by individuating which variables lead to the worse performances seen.

Bibliography

[1] A. A. Alves, Jr. *et al.* [LHCb Collaboration], "The LHCb Detector at the LHC," JINST **3** (2008) S08005. doi:10.1088/1748-0221/3/08/S08005

[2] L. Evans and P. Bryant, JINST **3**, S08001 (2008). doi:10.1088/1748-0221/3/08/S08001

[3] R. Aaij *et al.* [LHCb Collaboration], "First observation and measurement of the branching fraction for the decay $B_s^0 \to D_s^{*\mp} K^{\pm}$," JHEP **1506** (2015) 130 doi:10.1007/JHEP06(2015)130 [arXiv:1503.09086 [hep-ex]].

[4] F. Pedregosa *et al.*, "Scikit-learn: Machine Learning in Python," J. Machine Learning Res. **12** (2011) 2825 [arXiv:1201.0490 [cs.LG]].

[5] A. Hocker *et al.*, "TMVA - Toolkit for Multivariate Data Analysis," PoS ACAT (2007) 040 [physics/0703039 [PHYSICS]].

[6] S. Meloni *et al.*, "SMVA: Scikit-learn for Multivariate Analysis," LHCb internal note (2016) 039 LHCb-INT-2016-039.